Reconfigurable Microprocessors

Lih Wen Koh

05s1 COMP4211 presentation

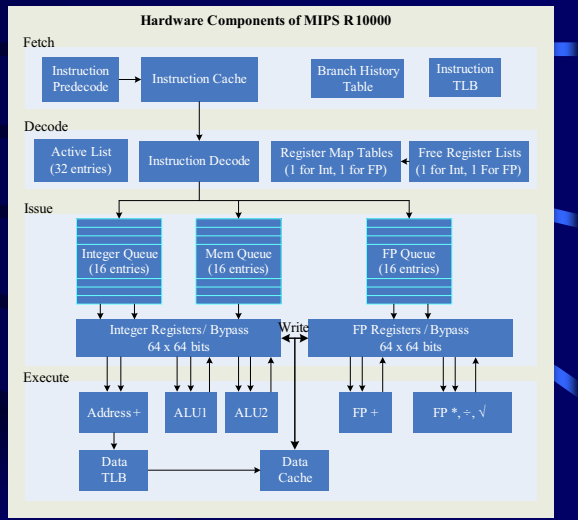
18 May 2005

Presentation Overview

- Current Research Direction
- Related Work
- Experiments
- What Next?

Current Research Direction

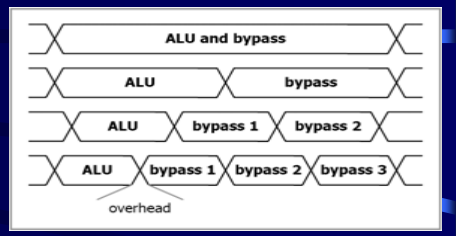
- Wide superscalar, out-of-order execution processor core
- Exploits ILP
- But true data dependencies are inherent in application programs
- MIPS R10k, NetBurst, AMD etc. use *bypass network* to forward just-computed result → allow back-to-back issue of dependent instructions
- Complexity of bypass network grows quadratic w.r.t. issue width



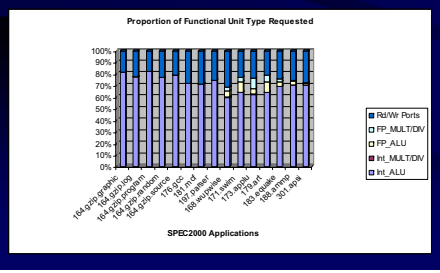
[Yeager96]

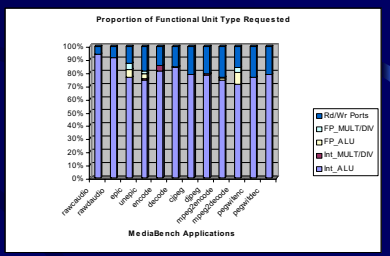
Current Research Direction

- Observation 1: Multi-cycle broadcast
  - Wire delays – accounted for in Intel NetBurst
  - Allows higher processor clock frequency at the cost of reduced IPC
- Observation 2: FP execution unit is idle most of the time, even in FP-intensive applications (5-10%)



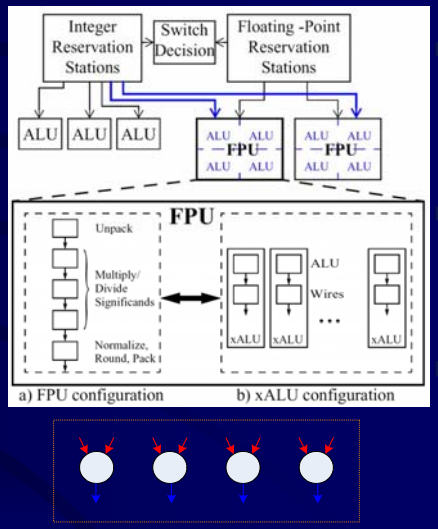
[Sassone04]





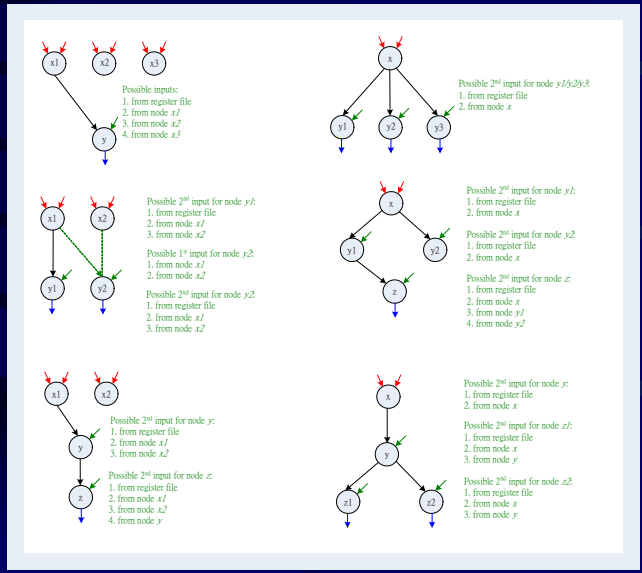
Literature Survey

- [Epalza04]: Dynamic Allocation of Functional Units in Superscalar Processors
  - Switch the execution mode of idle floating-point units to four additional integer ALUs
  - Addition of bypass networks add 1 cycle latency to the modified FPU.
  - 19% speedup for SPECint2000
  - 3.5% speedup for SPECfp2000
  - Issues:
    - Need to improve control for mode switching



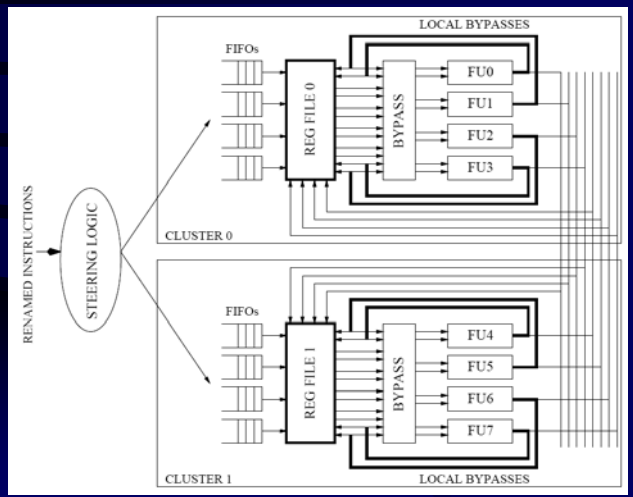
Plans

- Other patterns:



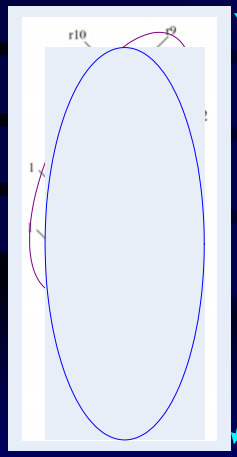
Related Work

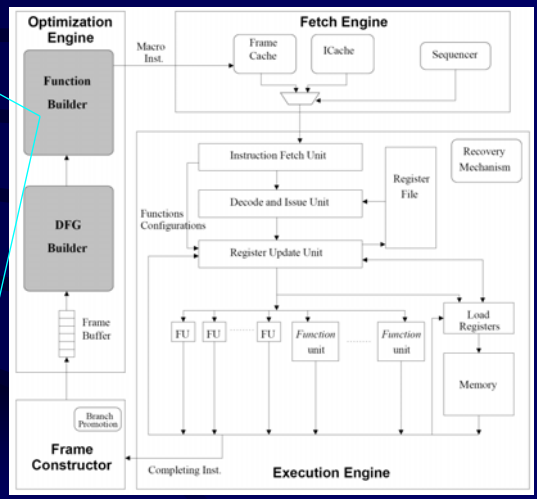
- [Palacharla97] Dependence-based (FIFO queues + clustered execution units)



Related Work

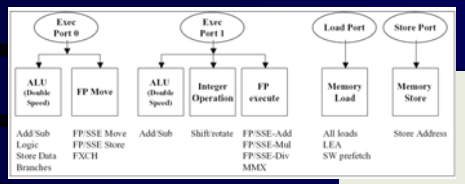
- Extension to rePLay framework [Yehia04]



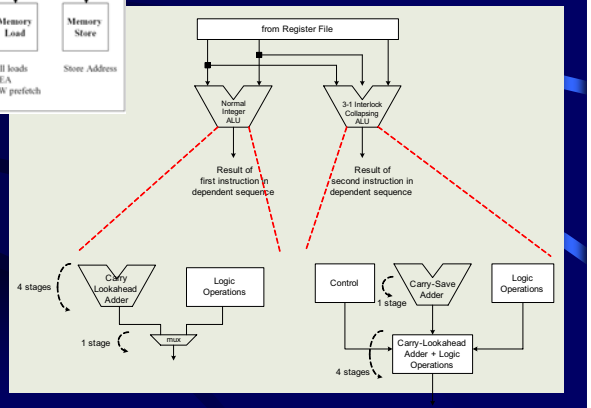


Experiment : Chaining pairs of dependent instructions

- [Intel01] Double-speed ALUs

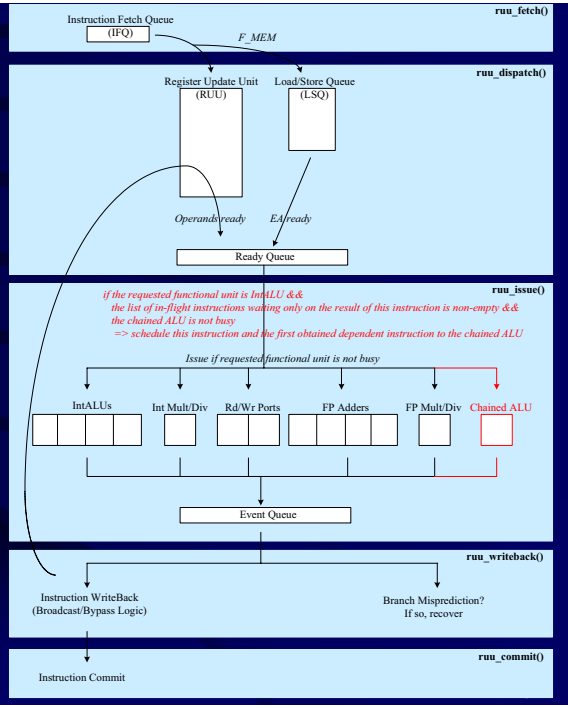


- [Vassiliadis96] 3-1 Interlock Collapsing ALUs



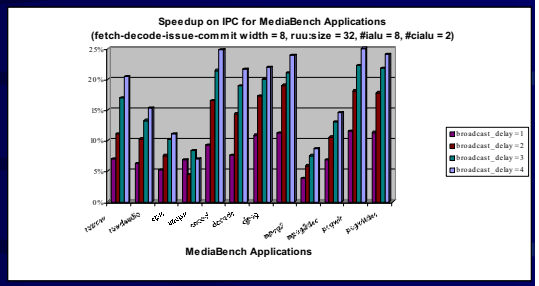
Experiment : Chaining pairs of dependent instructions

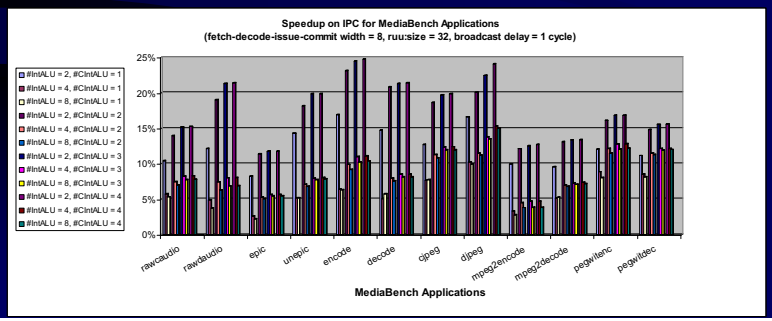
- Modifications to *sim-outorder* for SimpleScalar PISA.



Experiment : Chaining pairs of dependent instructions

- 2 CIALUs sufficient
- IPC improvement of ~8%, solely due to savings of broadcast cycles
- Reduces utlization of IALUs by ~50%
- Reduces up to 45% of queue entries waiting for result
- Up to 25% speedup as broadcast cycles = 4





What Next?

- Chaining sequence of 3 dependent instructions, other patterns out of the 80.
- Architectural impact of adding chained units
  - complexity of local bypass network etc.
- Replace chained units by xALUs converted from the CSA trees in a FP multiply/divide unit
  - Need to explore the hardware circuits of FP multiply/divide
- Develop an adaptive configuration scheme – to best match the interconnections of the swappable xALUs to the patterns of in-flight instructions.
  - Need to determine the most frequent subset of patterns

References

[Vassiliadis96] *High-Performance 3-1 Interlock Collapsing ALUs*. James Phillips and Stamatis Vassiliadis.

[Yeager96] *The MIPS R10000 Superscalar Microprocessor*. Kenneth C. Yeager. IEEE Micro 1996.

[Palacharla97] Subbarao Palacharla, Norman P. Jouppi, J.E. Smith. *Complexity-Effective Superscalar Processor*. ISCA 1997.

[Intel01] *The Microarchitecture of the Pentium® 4 Processor*. Glenn Hinton, Dave Sager, Mike Upton, Darrell Boggs, Doug Carmean, Alan Kyker, Patrice Roussel Intel Technology Journal Q1. 2001.

[Epalza04] *Dynamic Reallocation of Functional Units In Superscalar Processors*. Marc Epalza, Paolo Ienne, Daniel Mlynek. In the 9th Asia-Pacific Computer Systems Architecture Conference (ACSAC), 2004.

[Yehia04] *From Sequences of Dependent Instructions to Functions: A Complexity-Effective Approach for Improving Performance without ILP or Speculation*. Sami Yehia and Olivier Temam.

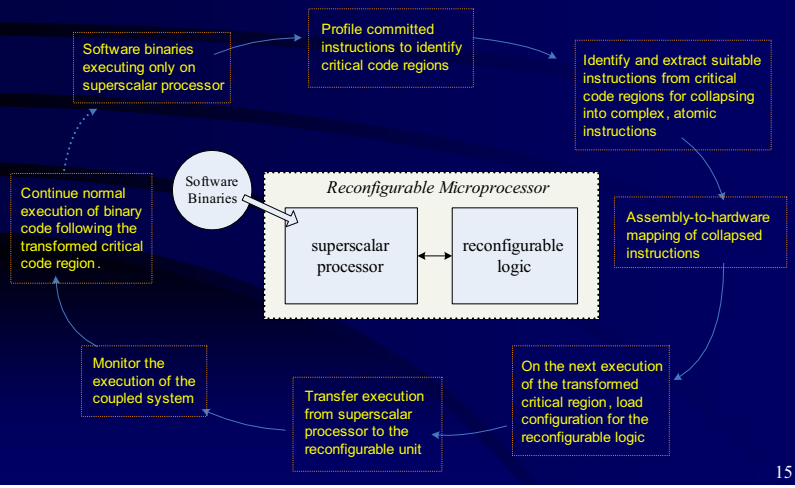
[Sassone04] *Multicycle Broadcast Bypass: Too Readily Overlooked.* Peter G. Sassone and D. Scott Wills, Proceedings of the Workshop on Complexity-Effective Design (WCED), May 2004.

Thank You

Overview of Research Topic

- Goal of this research:

  "investigate the feasibility and potential benefit of effective, automated **runtime compilation** and execution of **software binaries** on **reconfigurable microprocessors**"



Motivations

- Improved execution performance by exploiting parallelism and redundancy in hardware.
- Adaptation of hardware resources based on the dynamic behaviour of programs.
- Availability of runtime profile allows exploitation of runtime optimizations otherwise difficult to exploit at compile time.
- Compilation at the binary level allows execution of legacy software binaries.
- Runtime compilation allows transparent migration of software code to hardware.